Intersection-Validation: A Method for Evaluating Structure Learning without Ground Truth

Jussi Viinikka **Ralf Eggeling** **Mikko Koivisto**
Department of Computer Science, University of Helsinki, Finland
{jussivii,eggeling,mkhkoivi}@cs.helsinki.fi

Abstract

To compare learning algorithms that differ by the adopted statistical paradigm, model class, or search heuristic, it is common to evaluate the performance on training data of varying size. Measuring the performance is straightforward if the data are generated from a known model, the *ground truth*. However, when the study concerns real-world data, the current methodology is limited to estimating predictive performance, typically by cross-validation. This work introduces a method to compare algorithms' ability to learn the model structure, assuming no ground truth is given. The idea is to identify a partial structure on which the algorithms agree, and measure the performance in relation to that structure on subsamples of the data. The method is instantiated to structure learning in Bayesian networks, measuring the performance by the structural Hamming distance. It is tested using benchmark ground truth networks and algorithms that maximize various scoring functions. The results show that the method can produce evaluation outcomes that are close to those one would obtain if the ground truth was available.

1 INTRODUCTION

Evaluating the *statistical efficiency* of an estimator for some *parameter* of interest is a fundamental task when developing new estimators or applying existing ones. Two primary examples of parameter estimation are density estimation and structure learning in graphical models. Research in machine learning often proposes estimators that take the form of an *algorithm*, which takes data as input and produces some estimate as output. Due to the analytic intractability of the statistical model or the computational procedure, evaluating the statistical efficiency usually relies on *empirical studies*, in which the performance of one or several algorithms is measured in relation to synthetic or real-world data.

If one relies on synthetic data generated from a fixed model, the so-called *ground truth*, then evaluating the statistical efficiency is straightforward, aside from possible computational issues: One samples data sets of varying sizes, applies the estimator, and compares the obtained estimate to the parameter of interest. The main methodological questions then concern the choice of the ground truth and the choice of the performance measures. For example, in the context of learning Bayesian networks—the focus of this paper—it has been common to generate data from some benchmark networks, like Alarm [Beinlich et al., 1989], and study how well the learned models recover the data generating probability *distribution* or the network *structure* [Liu et al., 2012]. For the former, a standard metric is the *cross entropy* (CE) between the learned distribution and the ground truth [Heckerman et al., 1995]; for the latter, the *structural Hamming distance* (SHD) [Tsamardinos et al., 2006] has become popular.

Evaluating the efficiency in relation to real-world data is less straightforward. As the data generating model is not known, or it may not even exist in the form of the statistical model underlying the estimator, there is no direct way to compare the obtained estimate to the parameter of interest. There are indirect ways, however, provided that the parameter is "intimately related" to observable data. The distribution of data points is an example of such a parameter; the *cross-validation* method enables approximating CE (or any related metric). For other parameters, like the model structure, the present authors are not aware of any corresponding method. For example, Bayesian networks learned from benchmark data sets have not been eval-

Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

Table 1: Evaluation measures and methods for the statistical efficiency of learning Bayesian networks.

		Parameter (learning target)	
		Distribution	*Structure*
Ground truth	*Known*	Cross entropy, CE	Structural Hamming distance, SHD
	Unknown	Approximate CE by cross-validation	Approximate SHD by InterVal (*this paper*)

uated by approximating SHD, but either based on an implied estimated distribution (thus a different parameter) or using a given scoring function, like the BIC score, in special settings where all the compared algorithms aim to maximize the same scoring function.

To summarize, we currently lack a method for evaluating the statistical efficiency of structure learning in relation to real-world data and structural metrics; see Table 1 for a summary of the state of the art in learning Bayesian networks. Can we find a method that is to structure learning as cross-validation is to learning the distribution?

In this paper, we put forward such a method, we call *intersection-validation* (InterVal). The method applies to a setting where the interest is evaluating the relative performance of two or more algorithms. The idea is to first identify a partial structure on which the different algorithms agree when given the full data set, and then measure the distance between that partial structure and the structures the algorithms learn on subsamples of the data. The hope is that the partial structure mostly consists of structural features that are "easiest to learn" and part of the unknown ground truth, yet distinguishing the algorithms on smaller data sets.

We instantiate the InterVal method for evaluating the statistical efficiency of structure learning in Bayesian networks. Specifically, we present a definition of partial structure in that context. We also investigate empirically to what extent and under what conditions the method is able to produce essentially the same evaluation outcomes one could obtain if the ground truth was known. However, our study makes only the first steps in this direction, leaving several questions to be answered by future work.

The remainder of this paper is organized as follows. We review some basic concepts and notation of Bayesian networks in Section 2. Section 3 presents the InterVal method. In Section 4 we investigate the performance of the method in various scenarios based on experimental results. We conclude by discussing the prospects and limitations of the method in Section 5.

2 PRELIMINARIES

All the graphs we consider are directed and simple. Let $G = (V, E)$ be a graph with node set V and edge set $E \subseteq V \times V$. We denote a node pair (u, v) by uv and say that its *type* in G (or, in E) is *bidirected*, *forward*, *backward*, or *nonadjacent* if, respectively, both uv and vu, only uv, only vu, or neither belongs to E.

Suppose G contains no directed cycles, i.e., it is a *directed acyclic graph*, DAG. With each node $v \in V$ associate a random variable X_v. Let p be a probability distribution over the $|V|$ variables. The pair (G, p) is a *Bayesian network*, BN, if the joint distribution factorizes as $\prod_v p\big(X_v \,|\, (X_u)_{u \in G_v}\big)$, where $G_v = \{u : uv \in E\}$ is the set of *parents* of v in G.

Two DAGs G and G' on the same node set V are *equivalent* if they host the same set of distributions, i.e., (G, p) is a BN if and only if (G', p) is a BN. This holds exactly when the two DAGs have the same skeleton (i.e., the same set of nonadjacent node pairs) and v-structures (i.e., triplets of nodes u, u', v such that uu' is nonadjacent in G while $u, u' \in G_v$). The equivalence class of G is represented by the *completed partial DAG*, CPDAG, whose node set is V and the edge set is the union of the edge sets of the equivalent DAGs.

The *structural Hamming distance* between two CPDAGs C and C', denoted by $\mathrm{SHD}(C, C')$, is the number of node pairs whose types are different in the two graphs [Tsamardinos et al., 2006].

The *cross entropy* between two probability distributions p and q, denoted by $\mathrm{CE}(p, q)$, is the expected value of $-\ln q(X)$ under p. For discrete distributions we thus have that $\mathrm{CE}(p, q) = -\sum_x p(x) \ln q(x)$. A practical alternative to exact evaluation of the metric is to take the average of $-\ln q(X)$ over a large number of independent draws X from p.

In the context of this paper a *data set* is a sequence of data points $X^1, X^2, \ldots, X^N$, each of which can be viewed as a draw from some BN (G, p), i.e., from p.

3 INTERSECTION-VALIDATION

We consider K *learning algorithms* $A_1, A_2, \ldots, A_K$. Each A_i takes a data set D as input and returns a CPDAG $A_i(D)$ as the output. Our interest is in estimating how fast, as a function of data size, each algorithm's output approaches the ground truth CPDAG,

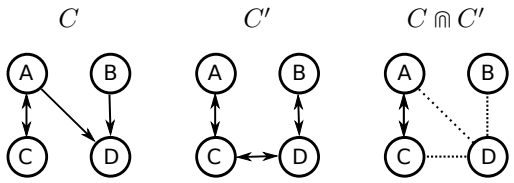

Figure 1: Two CPDAGs and their agreement graph. Dashed lines connect excluded node pairs.

C_*, that we assume exists but is unknown. Specifically, we wish to rank the algorithms based on the structural Hamming distance $\mathrm{SHD}(A_i(D), C_*)$, varying the data size, but relying on a single given data set D_0.

The idea of the *InterVal* method, which we formalize in the remainder of this section, is as follows. We first extract what we call an *agreement graph*, A_0, that represents the common ground of the K CPDAGs $A_i(D_0)$ and serves as a proxy of the ground truth. Then we measure the distance between each $A_i(D)$ and A_0, for varying subsamples D of D_0, by an appropriate variant of the structural Hamming distance, and base the ranking of the algorithms on these distances.

3.1 The Agreement Graph

For a collection of graphs, the agreement graph will be, in essence, the set of node pairs whose type is the same in every graph in the collection. To treat such objects more formally, we use a relaxed notion of graph: a *partial graph* on a set of node pairs S is a pair (S, E) where $E \subseteq S$. Note that an ordinary graph on V is obtained as a special case with $S = V \times V$. We define an intersection operation on partial graphs:

Definition 1. The *strict intersection* of two partial graphs $P = (S, E)$ and $P' = (S', E')$ is the partial graph $P \Cap P' = (I, E \cap E' \cap I)$, where $I \subseteq S \cap S'$ is the set of node pairs whose type is the same in P and P'.

Strict intersection differs from the ordinary intersection operation of graphs where we always have $I = S \cap S'$. The following basic fact allows us to talk about the intersection of multiple partial graphs without specifying the order of pairwise operations.

Proposition 1. *The strict intersection operation is commutative and associative.*

In what follows, we only consider strict intersections of CPDAGs and refer to an intersection as the *agreement graph*. Figure 1 shows an illustrated example.

3.2 The Partial Hamming Distance

We cannot use the structural Hamming distance for measuring the distance between a learned CPDAG and the agreement graph because the latter argument is generally not a CPDAG. Therefore we consider the following straightforward extension:

Definition 2. The *partial Hamming distance* between two partial graphs $P = (S, E)$ and $P' = (S, E')$, denoted by $\mathrm{PHD}(P, P')$, is the number of node pairs in S whose types are different in P and P'.

It is easy to show that PHD is a metric.

Proposition 2. *The partial Hamming distance is a metric in the set of partial graphs on a fixed set of node pairs.*

Because we use PHD to measure the distance between a CPDAG and a partial graph, we first project the CPDAGs to the set of node pairs of the partial graph.

Definition 3. The *projection* of a CPDAG (V, E) to a set of node pairs S is the partial graph $(S \cap V \times V, S \cap E)$.

For a CPDAG C and a partial graph P on a set S we will write simply $\mathrm{PHD}(C, P)$ for $\mathrm{PHD}(C', P)$ where C' is the projection of C to S.

3.3 Estimation by Subsampling

The following method estimates the ranking of the given learning algorithms for a given sample size $s < |D_0|$. The method has one user parameter, the number of subsamples r; in our experiments we set $r = 10$.

1. Let A_0 be the agreement graph of $\{A_i(D_0)\}_{i=1}^{K}$.
2. For $t = 1, 2, \ldots, r$, construct a D_t by sampling s data points without replacement from D_0.
3. Compute the distance $d_{it} = \mathrm{PHD}(A_i(D_t), A_0)$ for all pairs of i and t.
4. Along with the empirical joint distribution $\mathcal{D}^r(s) = \{(d_{1t}, d_{2t}, \ldots, d_{Kt})\}_{t=1}^{r}$, report the per-algorithm sample means

$$\mu_i^r(s) = \frac{1}{r} \sum_{t=1}^{r} d_{it}$$

and possible other summary statistics.

In visualizations we typically plot the sample means $\mu_i^r(s)$ and the respective standard errors. However, depending on the precise formulation of the ranking problem, other statistics of the empirical distribution $\mathcal{D}^r(s)$ can be used.

3.4 Asymptotic Consistency

Consider a collection of Bayesian networks $\mathcal{B}$ on a fixed set of random variables X_v, $v \in V$. Let θ be

a mapping from $\mathcal{B}$ to some set R. We call a sequence of functions T_N that map a sequence of data points $X^1, X^2, \ldots, X^N$ to an element of R a *consistent estimator* of θ if for all $(G, p) \in \mathcal{B}$, the sequence T_N converges to $\theta(G, p)$ in probability as $N \to \infty$, assuming the data points are independent draws from p.

For example, a learning algorithm A is a consistent estimator of the CPDAG C_* of the data generating BN, or *consistent* for short, if the probability of the event $\{A(D_0) = C_*\}$ tends to one as the data size $N = |D_0|$ grows. It is known that, under mild conditions on the collection $\mathcal{B}$, an algorithm is consistent if it maximizes a well-behaving scoring function, such as the BIC score [Koller and Friedman, 2009, Thm. 18.2].

Let $\mu_i(s)$ be the expected value of $\mathrm{SHD}(A_i(D), C_*)$, where the data set D consists of s independent draws from the data generating BN. Now, the consistency of learning algorithms translates directly to the consistency of the InterVal estimators:

Theorem 3. *Let $A_1, A_2, \ldots, A_K$ be consistent learning algorithms. Then for all $i \in [K]$ and $s \in \mathbb{N}$ we have that $\mu_i^N(s)$ is a consistent estimator of $\mu_i(s)$.*

This result follows because the agreement graph A_0 converges to C_* and so $\mathrm{PHD}(\cdot, A_0)$ converges to $\mathrm{SHD}(\cdot, C_*)$; note that in $\mu_i^N(s)$ we could replace N by any number that grows with N. While this simple asymptotic result is of little use in practice, it summarizes the idea of the InterVal method and servers as a starting point of the quest for finite-sample guarantees.

4 EXPERIMENTAL RESULTS

In this section, we empirically compare intersection-validation with the other three methods for evaluating structure learning algorithms (Table 1).

For obtaining algorithms that produce different results on finite data while still being asymptotically consistent, we used exact score based learning with varying scoring functions: BDeu [Heckerman et al., 1995] with four different ESS values (0.01, 0.1, 1, 10), the BIC score [Schwarz, 1978], and fNML [Silander et al., 2010]. We computed globally optimal DAGs using GOBNILP [Cussens, 2011, Bartlett and Cussens, 2013]. The program takes as input either data files or pre-computed local scores. For the first case, when directly working with data, the program supports only BDeu with varying ESS as scoring function. Therefore for BIC and fNML we first calculated the local scores using the URLearning package (`http://urlearning.org`) [Yuan and Malone, 2013]. To limit the computational effort, for all of the six scoring functions we set the maximum indegree parameter to four. We estimated the parameters according to the posterior mean with the prior coinciding with structure learning; for BIC and fNML, we set the ESS to 5.0.

Table 2: **Benchmark networks.** MaxIn is the maximum indegree and Param the number of free parameters of the network.

Network	Nodes	Arcs	MaxIn	Param
Sachs	11	17	3	178
Insurance	27	52	3	984
Alarm	37	46	4	509

For benchmarking we used three classic fully specified Bayesian networks as ground-truth: Alarm [Beinlich et al., 1989], Insurance [Binder et al., 1997], and Sachs [Sachs et al., 2005]; we obtained the networks from `www.bnlearn.com/bnrepository`. Due to the moderate numbers of variables and arcs (Table 2) these networks are suitable for structure learning using GOBNILP.

For each benchmark network, we generated data sets of varying size

$$N = 100 \times 2^n\,, \text{ with } n = 0, 1, \ldots, 8\,, \qquad (1)$$

and repeated the process ten times, obtaining $3 \times 9 \times 10 = 270$ data sets in total.

4.1 Performance of Established Methods

Before evaluating InterVal itself, let us first examine the performance of the three known evaluation methods of Table 1 on the chosen benchmark data sets in order to obtain a reference point for the comparisons in the later sections.

For each of the generated data sets, each of the six learning algorithms returned a Bayesian network, which we evaluated based on two distance measures: structural Hamming distance (SHD), which compares only the network structures, and cross-entropy (CE), which compares the entire distribution. In addition, we computed the mean log-predictive probability via ten-fold cross-validation (CV). For each benchmark network, algorithm, and sample size we averaged each performance measure over the ten generated data sets. The resulting learning curves for all data sets and evaluation methods are shown in Figure 2.

By visually inspecting the curves, we find that both ground-truth based evaluation metrics (SHD and CE) lead to different conclusions about the performance of the six learning methods relative to each other. Conversely, comparing CE and CV we find that the latter reproduces the curves of the former with remarkable

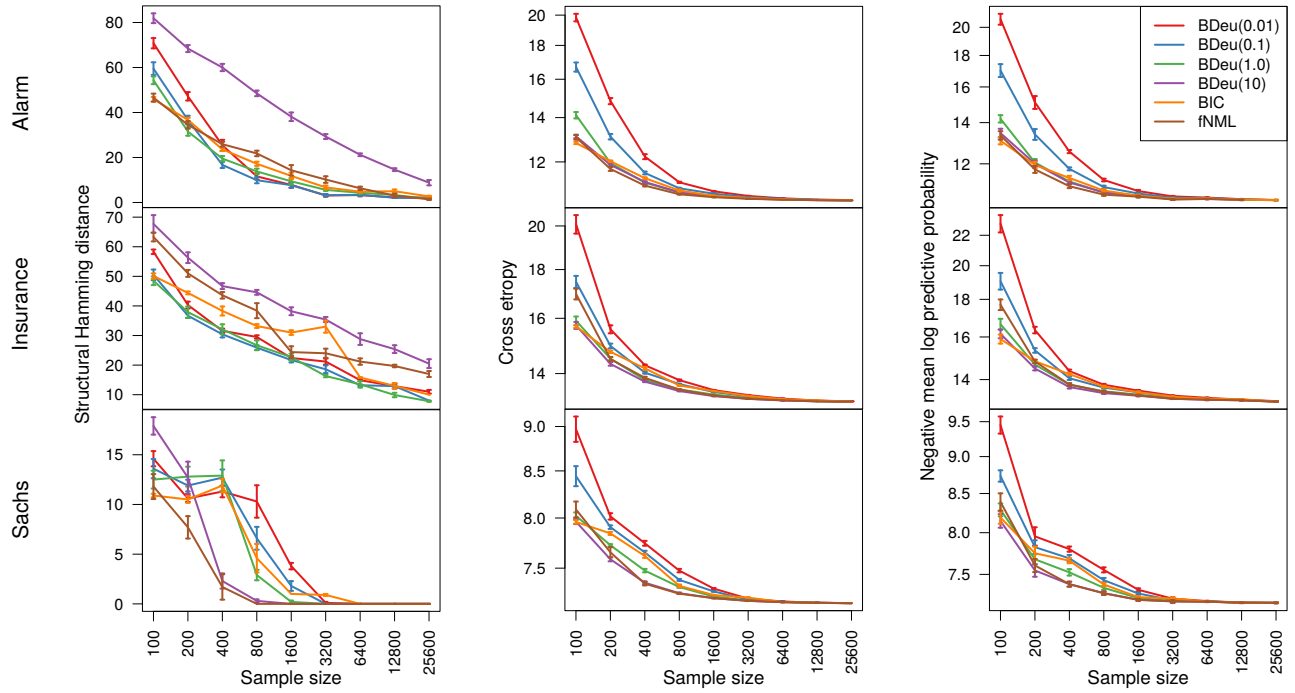

Figure 2: Evaluation of learning algorithms with existing methods for three exemplary data sets.

accuracy. We conclude that CV is a poor approximation of SHD, so a method that evaluates structure learning in absence of a ground truth is needed.

We also observe that SHD converges only in case of Sachs, where eventually all methods find the correct CPDAG. For Insurance and Alarm no learning method achieves this at a sample size of 25,600 data points yet. For all networks, CE and CV seem to converge already at around 12,800 data points, convergence of the distribution happens much faster than convergence of the structure. Finding the generating network structure (up to the equivalence class) is thus a harder problem than approximating the generating distribution.

One challenge in using learning curves, as in Figure 2, for comparing evaluation methods is that visual interpretation needs—to some degree—subjective judgment. To quantify differences between two evaluation methods more objectively, we took for a given data set the six mean performance measures for each of the two methods in comparison and computed the Pearson correlation among them. We show these correlations for the pairwise comparison of CE vs SHD and CE vs CV in Figure 3. Missing values (sample size 25,600 for all networks and two further sample sizes for Sachs) are caused by a complete agreement of algorithms for at least one method, rendering Pearson correlation undefined due to zero variance. We omit the correlation of SHD vs CV for brevity as it is virtually identical to that of SHD vs CE.

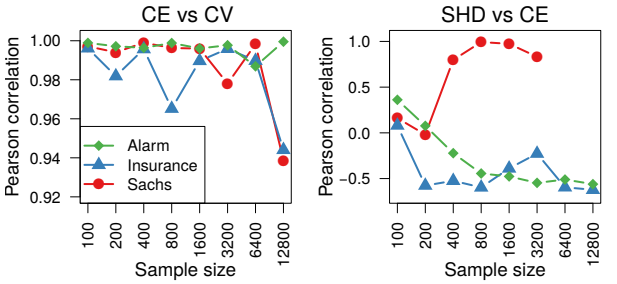

Figure 3: Correlation among evaluation methods.

The main conclusion from this study is that Pearson correlation is indeed suitable to quantify differences among learning curves numerically as it coincides with visual interpretation: For CE and CV, which are nearly indistinguishable by visual inspection, we obtain a correlation of 0.98 on average and of 0.94 in the worst case. Conversely, where is rankings obviously different, e.g., for comparing SHD and CE for Alarm and Insurance at all sample sizes, there is no positive correlation but rather a slight anti-correlation among the mean values.

4.2 Intersection-Validation: First Evaluation

Now we evaluate the InterVal method assuming the given data set is of size N, which we call the *intersection point*. For a moment we fix the intersection point

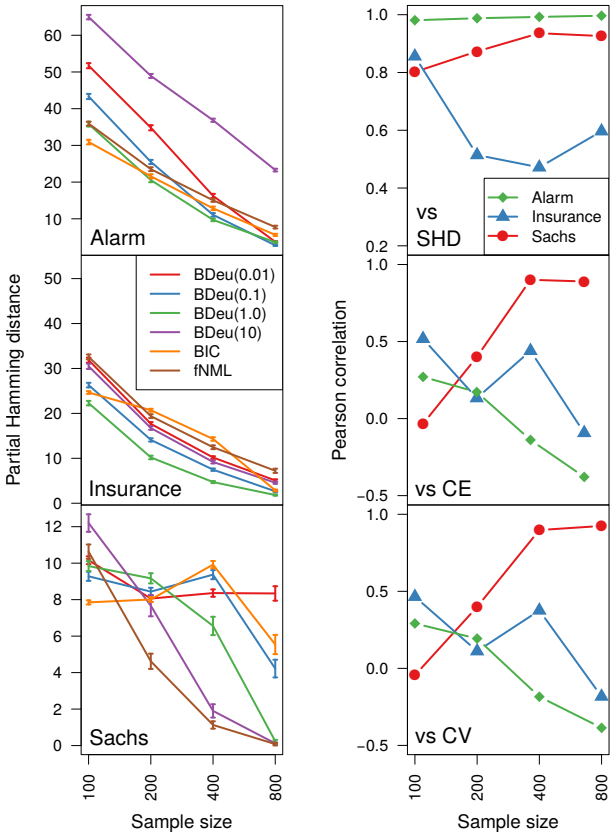

Figure 4: Intersection-validation for fixed intersection point 1,600. Left: Partial Hamming distance up to the intersection point 1600. Right: Correlation of PHD with other evaluation methods.

N to 1,600. We pick this value since ground-truth-based methods revealed that substantial learning happens already with that amount of data, but for none of the data sets all algorithms agree according to any evaluation method yet.

We computed agreement graphs from the results of the six structure learning algorithms at the intersection point. Next, we created from these maximal data sets ten subsamples according to the data set sizes specified in Eq. 1. Finally, we learned network structures from the subsampled data with all six methods and computed the partial Hamming distances between the learned networks and the agreement graph. We repeated this procedure for all ten data sets of size 1,600 and for all three generating networks and plot the averaged PHD for sample sizes $100, 200, 400,$ and 800. In addition, we computed for all data sets and sample sizes the correlation of PHD against the assessment of the other methods. The results are shown in Figure 4.

We observe that PHD reproduces the results of SHD in the case of Alarm almost perfectly with visually hardly distinguishable learning curves and a correlation of nearly 1. For Sachs, the correlations are a bit smaller, but that is not very surprising given the non-monotonic behavior of the SHD ground truth learning curves. For Insurance, however, the learning curves differ visually a bit and the correlation to SHD is smaller, ranging from 0.5 to 0.9, depending on the sample size. While it is still higher compared to the correlation of PHD to CE or CV, PHD as a proxy for SHD does not always perform as well as CV as a proxy of CE (cf. Figure 3). This might be due to the fact that the network structure converges slower than the distribution (Section 4.1) and is more prone to noise due to the lower resolution of the discrete state space.

4.3 Varying the Intersection Point

Next we study the performance of SHD for varying intersection points, simulating the property of real-world data sets to be of different size in relation to the (unknown) ground truth network. We repeated the computations described in Section 4.2 for all intersection points from 200 onwards according to Eq. 1 and all three generating networks. The resulting 24 plots of PHD learning curves are shown in the Supplement.

Figure 5 summarizes the results by displaying the correlation between PHD and SHD as a function of subsample size for all intersection points. We observe that InterVal approximates SHD well for most locations of the intersection point, but the performance varies.

For Alarm the decay is very small, even taking the intersection point of at sample size 200 achieves a correlation of 0.93 to the ground-truth based method at sample size 100. This generating network demonstrates that InterVal can work almost perfectly when the ground-truth SHD curves are sufficiently smooth and some parts of the network are learned correctly already at small sample sizes.

In the case of Insurance the correlation of PHD and SHD varies a lot. One explanation might be that, in relation to the size of the network, Insurance shows the slowest convergence w.r.t. SHD (Figure 2). Even at sample size 25,600 all algorithms have still an SHD of at least ten, whereas for Alarm all but BDeu(10) find almost the correct equivalence class. Another, explanation might be the rather unstable SHD learning curves under ground truth at relatively large sample sizes, e.g., two late crossings of BIC and fNML.

For Sachs, we need at least intersection point 800 to approximate the ground-truth based assessment accurately. A possible explanation is given by the SHD plot in Figure 2, where at sample size 400 four of the six algorithms still have an SHD of about 10, indicating that they have made not much progress in identifying the correct structure yet. As a consequence, the

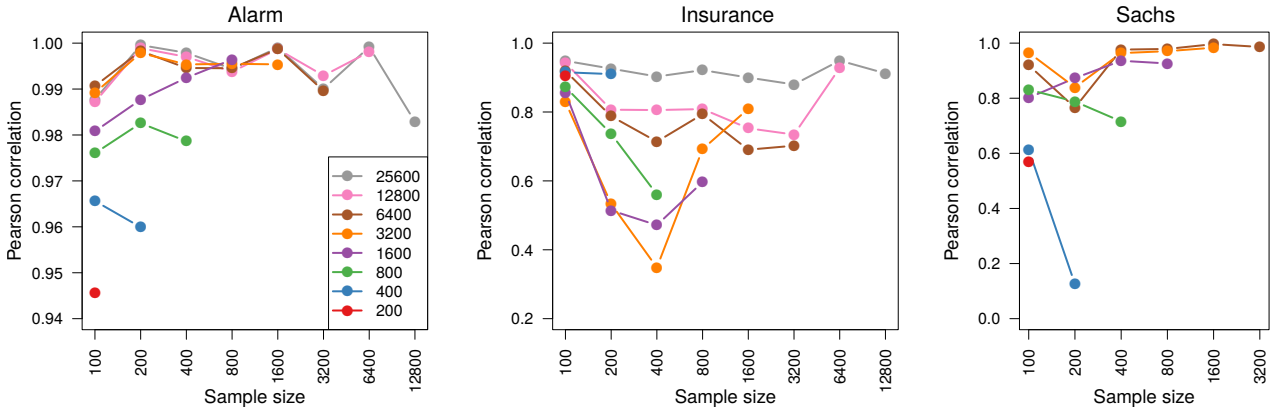

Figure 5: Correlation PHD vs SHD for different intersection points (legend) and sample sizes (x-axis).

agreement graph is at intersection point 400 hardly more than guesswork and InterVal cannot reproduce the SHD-based assessment for smaller sample sizes.

We conclude that the amount of available data at the intersection point matters to some degree, but it is not an equally important factor for all data sets and distributions. In addition, the stability of the SHD-learning curves under ground-truth also plays a role.

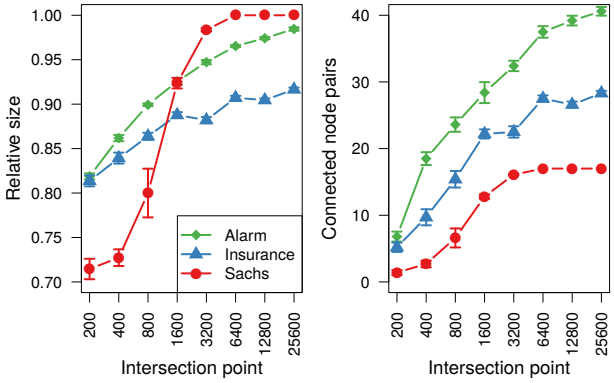

Figure 6: Properties of the agreement graphs.

4.4 Predicting the Performance

To further investigate under which conditions InterVal performs well, we next investigate the properties of the agreement graphs at different intersection points for the three generating networks. Figure 6 (left) shows the relative size of the agreement graph, that is, the number of included node pairs divided by number of total node pairs. While the relatives sizes are all larger than 0.7, it should be noted that a node-pair can also agree if none of the algorithm reports an edge among them. In particular, for sparse networks with many variables, such as Alarm, this type of node pairs may amount the vast majority.

Figure 7 shows for Sachs the ground-truth DAG and CPDAG as well as agreement graphs at different intersection points, which originate from the first of the ten data samples in the study. Interestingly, for this network there are no directed edges in the CPDAG as the original DAG has no unshielded v-structures. Comparing the agreement graphs to the ground truth CPDAG, we observe that most of the node-pairs that are not included in the agreement graphs (dashed lines) correspond to edges in the ground truth CPDAG. Comparing this to the other two generating networks (Supplement), we conclude that it is rather a peculiarity of the small number of variables, though. More important, edges that all algorithms agree upon are indeed in the ground truth and this observation also holds in the case of Alarm and Insurance.

Hence, we should also consider the absolute number of edges in the agreement graph, i.e., the number of *connected node pairs* (CNP) as informative statistic about the quality of the agreement graph. We plot the CNP in Figure 6 (right) and observe that at low sample sizes the CNP of the agreement graph is for all networks very small, indicating that initially most algorithms disagree completely. In particular, for Sachs, we have only 1.4 and 2.7 CNP in the agreement graph on average for intersection point 200 and 400, so it is hardly surprising that InterVal fails to reproduce SHD. Having no more than three non-empty node pairs as base for comparing algorithms appears to be too few. However, it is remarkable that agreement on as few as five CNP (intersection point 200 for Insurance) can be enough for a strong positive correlation between InterVal and SHD.

We also investigated whether the size of the agreement graph and/or the CNP can be used as a predictor for the accuracy of InterVal (Figure 8). There is a positive correlation for Sachs (0.801), and Alarm (0.858) between

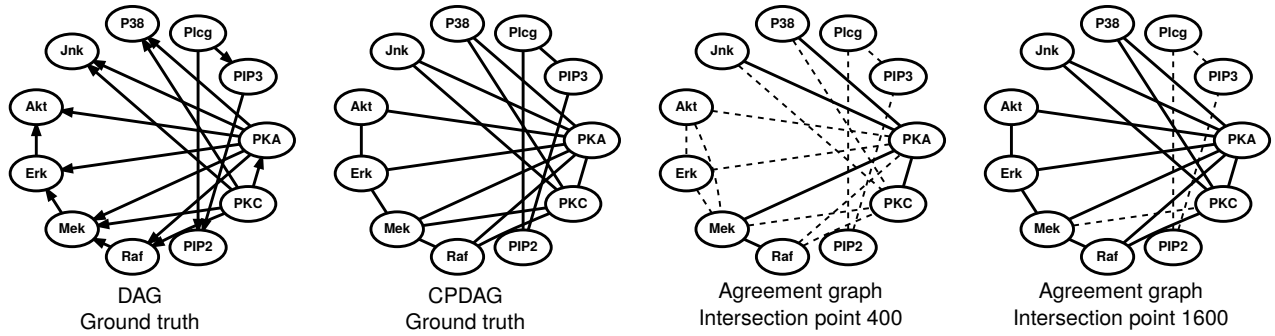

Figure 7: Comparison of Sachs ground truth network with agreement graphs from first data sample. Here we omit the arrow heads of bidirectional edges for simplifying the visualization.

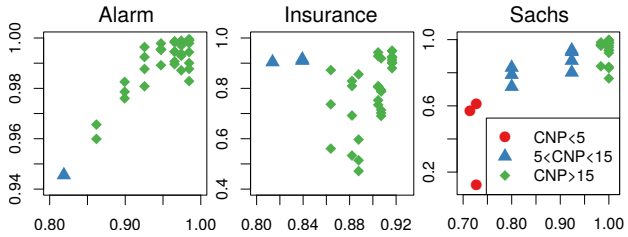

Figure 8: Accuracy of Intersection-Validation as correlation PHD to SHD (y-axis) vs size of agreement graph (x-axis) for all combinations of intersection point and sample size. Legend indicates number of connected node pairs (CNP) in the agreement graph.

the effectiveness of InterVal and the relative size of the agreement graph. But for Insurance the accuracy is relatively independent of the agreement graph size.

These results demonstrate that the relative size of the agreement graph and the absolute number of connected node pairs can be used as a predictor for the effectiveness of InterVal. While they cannot be the only determining factors, they allow to predict when InterVal should not be applied: When either the number of connected node pairs is less than five or the size of the intersection-graph amounts less than 80% of the total node-pairs, InterVal should not be used with much confidence. Conversely, having an agreement graph with a relative size greater than 0.9 allows—at least for the data sets in this study—a good approximation of SHD-based evaluation under known ground-truth.

5 CONCLUDING REMARKS

We have proposed a method that allows evaluating the quality of learned network structures even in the case that no ground-truth DAG is known. In contrast to evaluating the predictive performance via cross-validation, our InterVal method evaluates structural similarity among the network as learning target instead of the entire distribution.

We empirically demonstrated that the method yields conclusions about the performance of algorithms that resemble those obtained from measuring SHD if the ground-truth DAG was known. We also observed that these conclusions can differ dramatically from those that are obtained by evaluating the learned distribution. While the presented studies concern the comparison of six algorithms, InterVal can also be applied to the simpler pairwise comparison (Supplement).

In addition, we studied the question of whether the accuracy of InterVal can be predicted purely from what can be observed. We found that the relative size of the agreement graph as well as the number of connected node pairs it in give a reliable forecast on the accuracy.

One limitation of the InterVal method is that it cannot make a statement about the performance of algorithms on the entire given data set, but only on smaller subsamples. However, cross-validation has essentially the same property: 10-fold cross-validation evaluates the distribution at 90% of the given sample size.

We believe the proposed method has high potential and warrants future research in several directions. One is to seek rigorous finite-sample accuracy guarantees under practical assumptions. Another direction is to both expand the experiments in the variety of ground truth Bayesian networks and in the considered algorithms. It would be also interesting to study how well InterVal works with structural metrics other than SHD [de Jongh and Druzdzel, 2009, Peters and Bühlmann, 2015].

Acknowledgements

The authors thank the anonymous reviewers for valuable suggestions that helped to improve the presentation. This work was supported in part by the Academy of Finland, Grant 276864.

References

M. Bartlett and J. Cussens. Advances in Bayesian network learning using integer programming. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)*, pages 182–191. AUAI press, 2013.

I. Beinlich, H. Suermont, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, pages 247–256, 1989.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2-3):213–244, 1997.

J. Cussens. Bayesian network learning with cutting planes. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 153–160. AUAI press, 2011.

M. de Jongh and M. Druzdzel. A comparison of structural distance measures for causal Bayesian network models. *Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science series*, 443–456, 2009.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Z. Liu, B. Malone, and C. Yuan. Empirical evaluation of scoring functions for Bayesian network model selection. *BMC Bioinformatics*, 13(Suppl 15):S14, 2012.

J. Peters and P. Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Computation*, 27(3):771–799, 2015.

K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308: 523–529, 2005.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 2:461–464, 1978.

T. Silander, T. Roos, and P. Myllymäki. Learning locally minimax optimal Bayesian networks. *Int. J. Approx. Reasoning*, 51(5):544–557, 2010.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

C. Yuan and B. Malone. Learning optimal Bayesian networks: A shortest path perspective. *Journal of Artificial Intelligence Research (JAIR)*, 48:23–65, 2013.